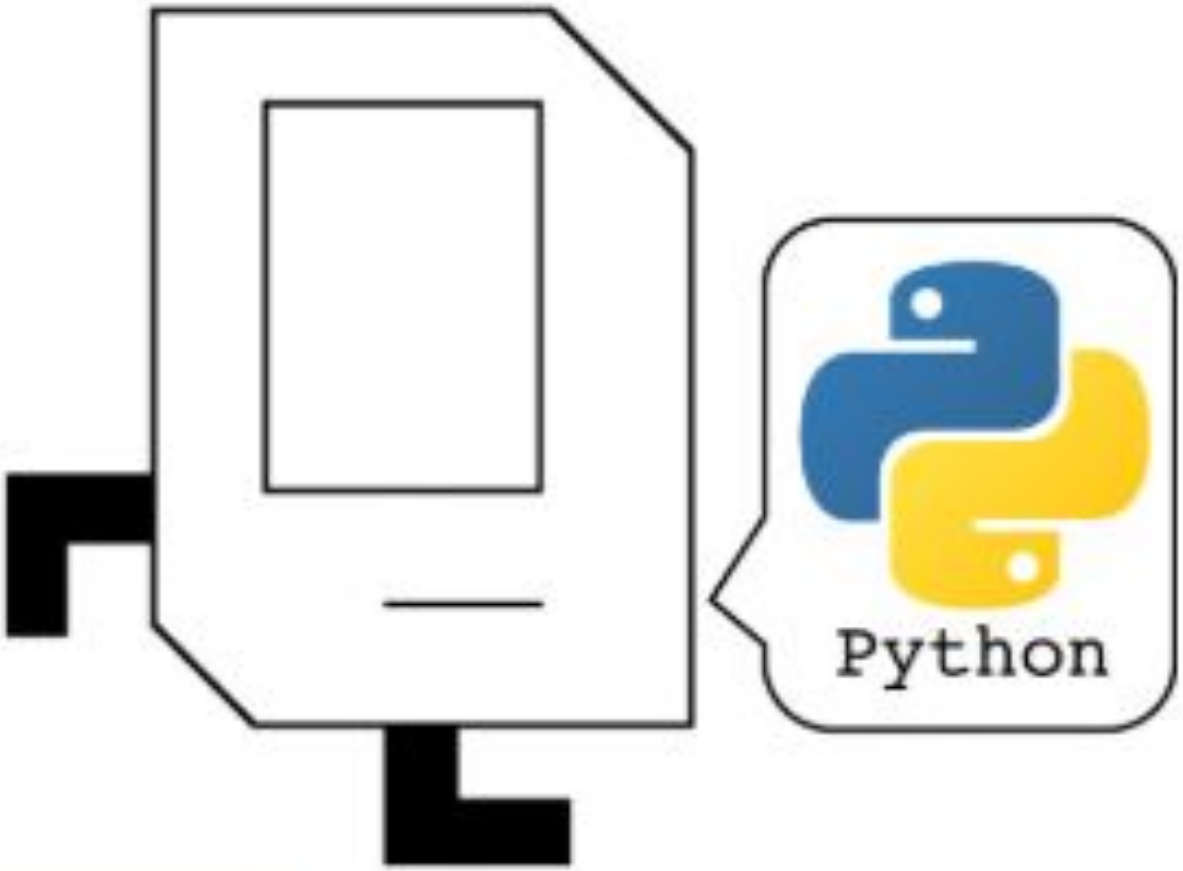
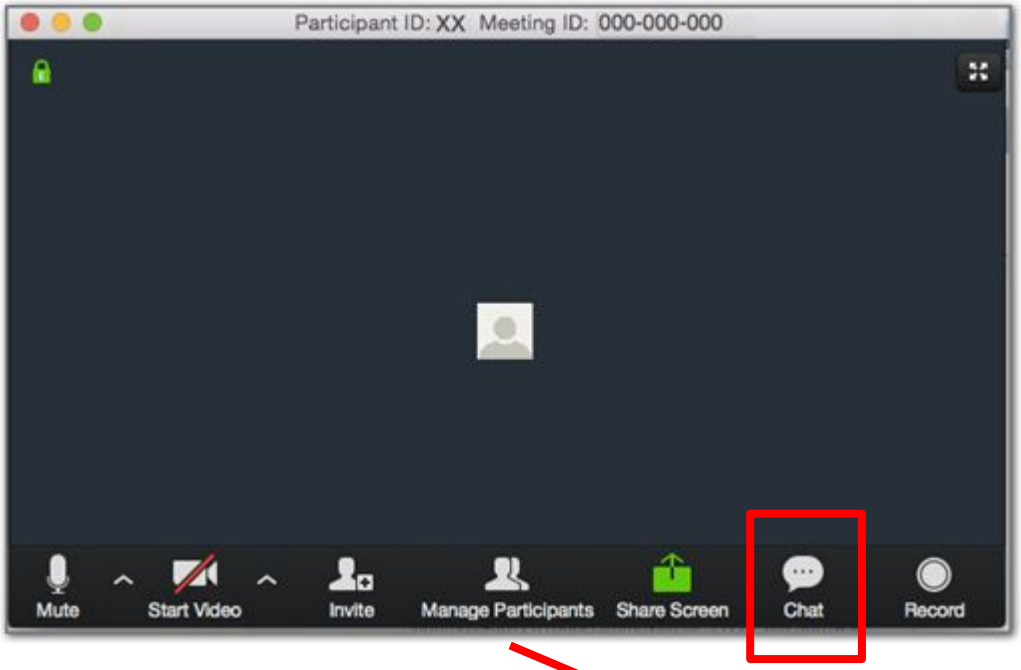
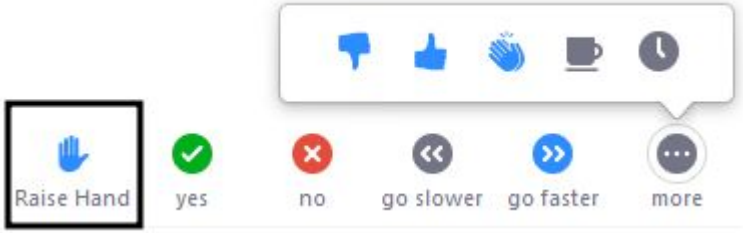
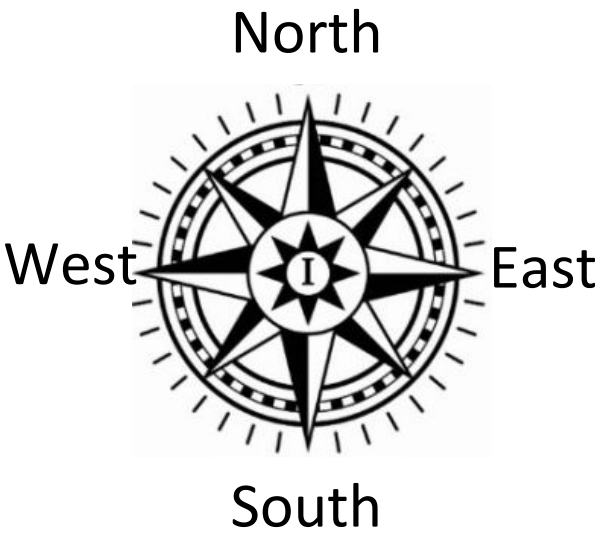

Control Flow

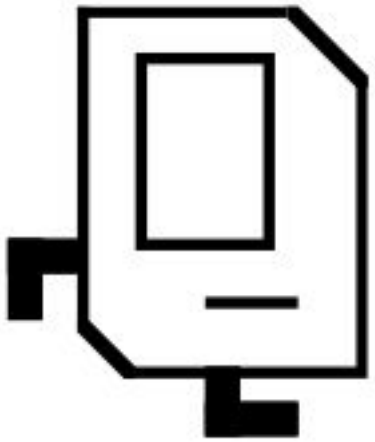

Python


Participant ID: XX Meeting ID: 000-000-000
Mute
Start Video
Invite
Manage Participants
Share Screen
Chat
Record


Raise Hand
yes
no
go slower
go faster
more


"Streets" run East/West
"Avenues" run North/South
3
2
1


North
West
East
South



```
move

turnLeft

putBeeper

pickBeeper
```

```
def main():
    move_to_newspaper()
    pick_beeper()


def move_to_newspaper():
    move()
    move()
    turn_left()
    turn_left()
    turn_left()
    move()
    turn_left()
    move()
```

```
def turn_right():
    turn_left()
    turn_left()
    turn_left()
```

```
def turn_right():
    for i in range(3):
        turn_left()
```

```
from karel.stanfordkarel import *

"""
File: collect_newspaper_karel.py
-------------------------------
Karel picks beeper in front of his house.
"""


def main():
    pick_beeper()
    move_to_start()


def turn_right():
    for i in range(3):
        turn_left()


def move_to_start():
    # turn around
    turn_left()
    turn_left()
    # move back to start
    for i in range(3):
        move()
    turn_right()
    move()
    # reorient to face right
    turn_right()


# There is no need to edit code beyond this point
if __name__ == "__main__":
    run_karel_program()
```

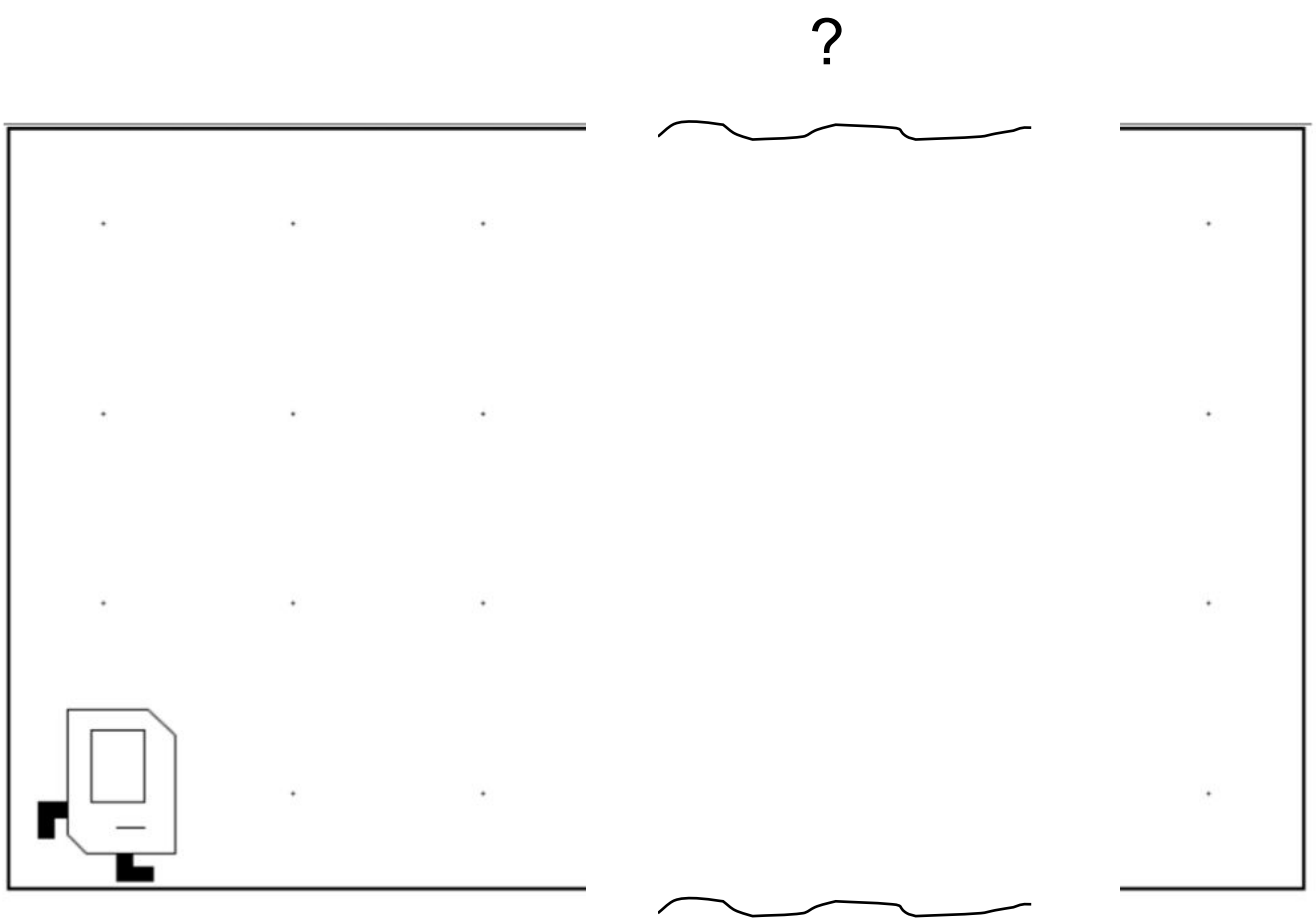
Multi-line comment

Main function

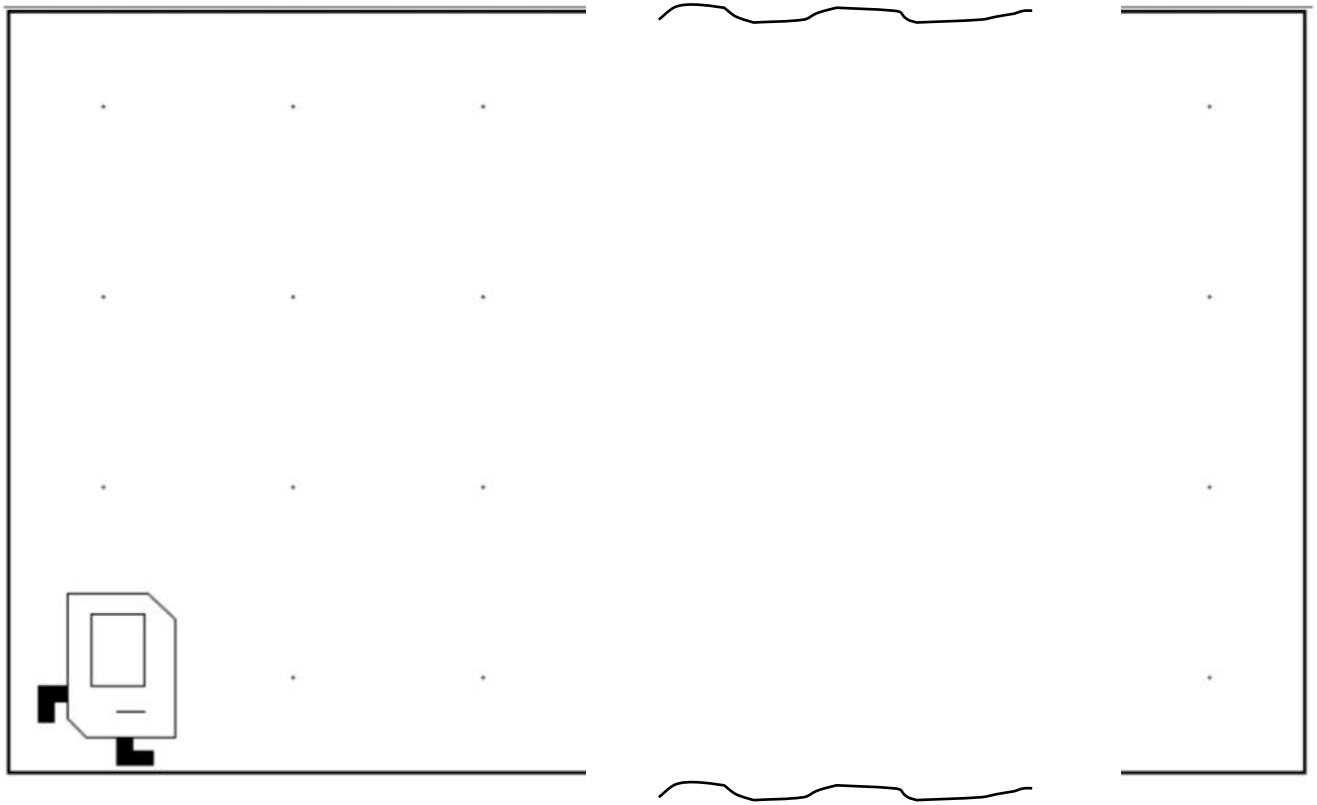
Our function

Single-line comment

Fill a street with beepers in a world of any size.

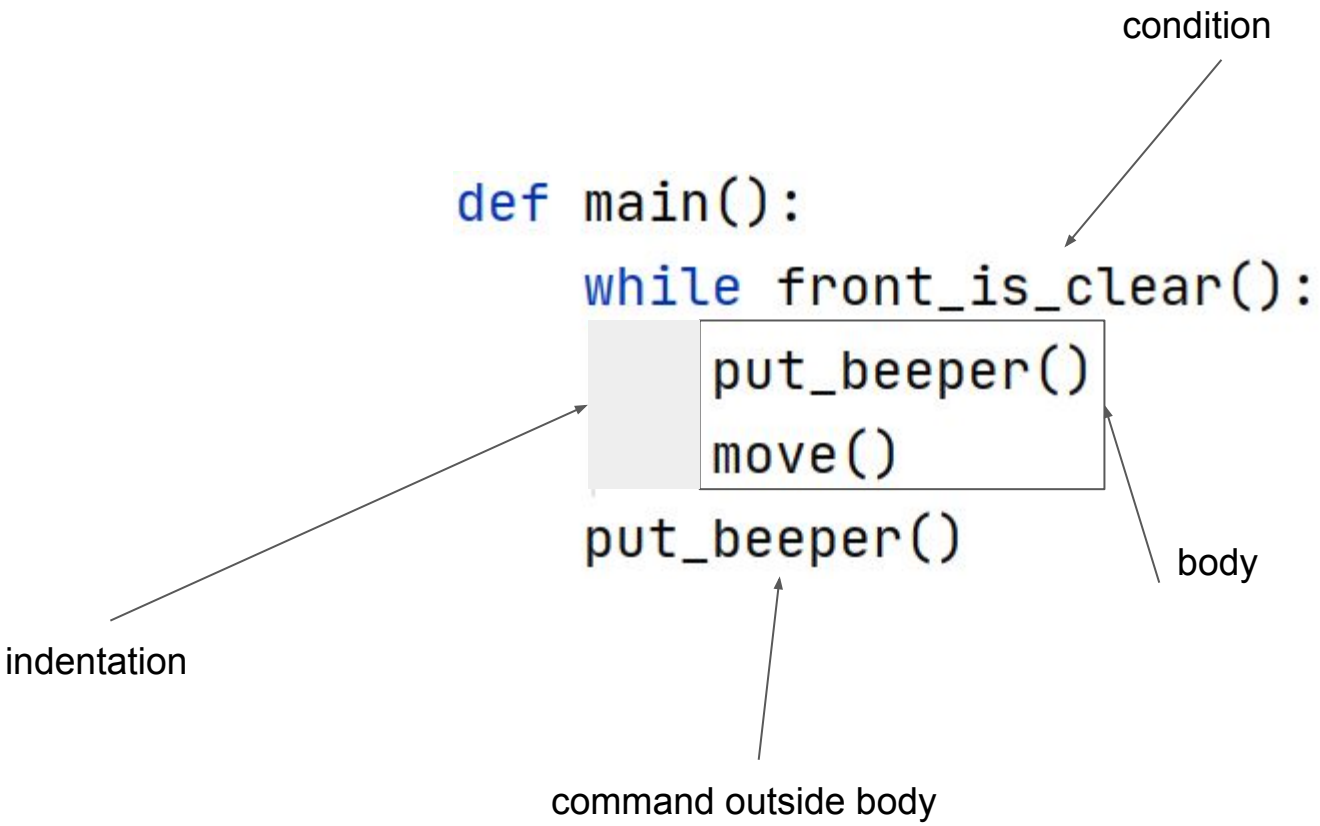


?

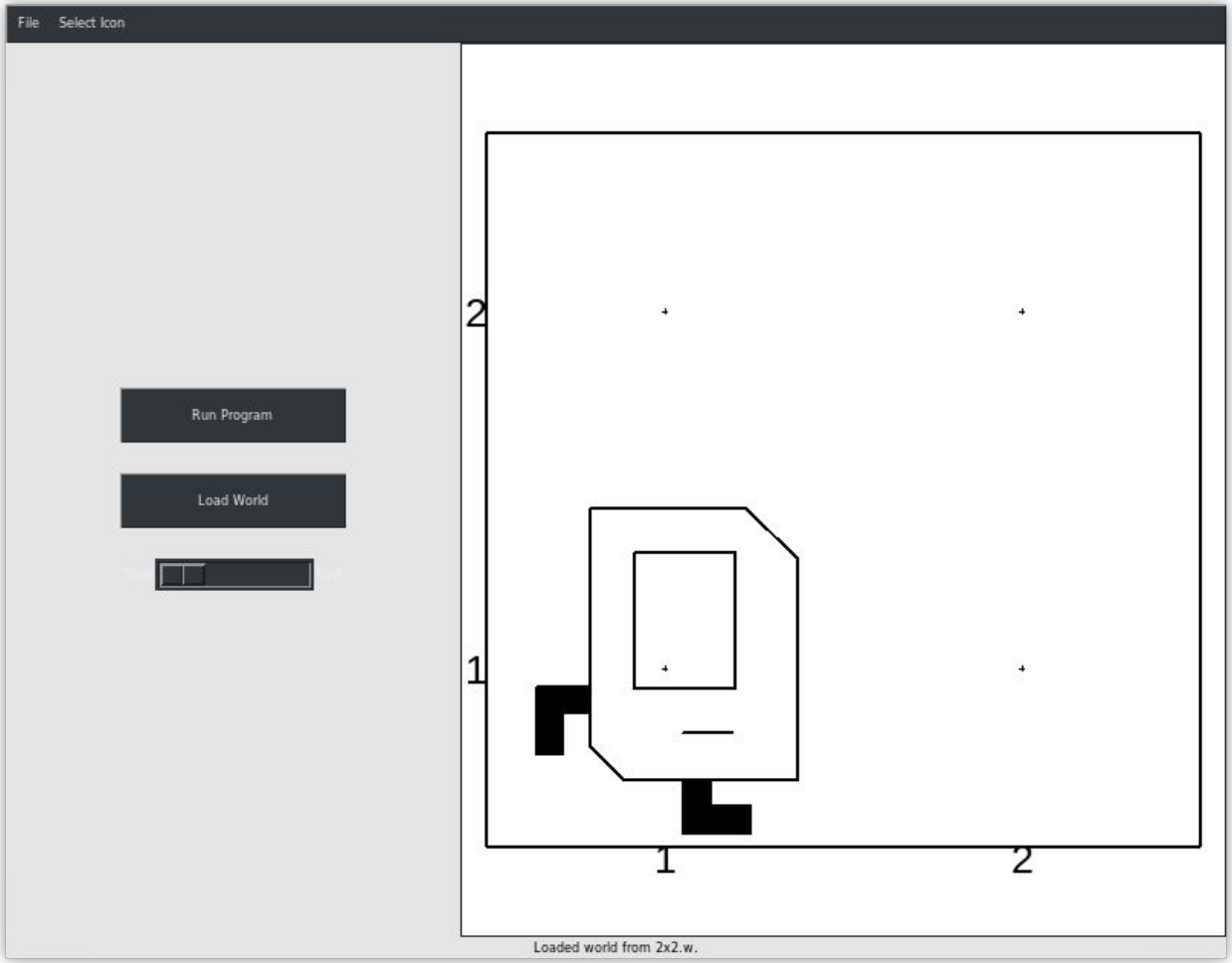


```
def main():
    for i in range(?):
        put_beeper()
        move()
```

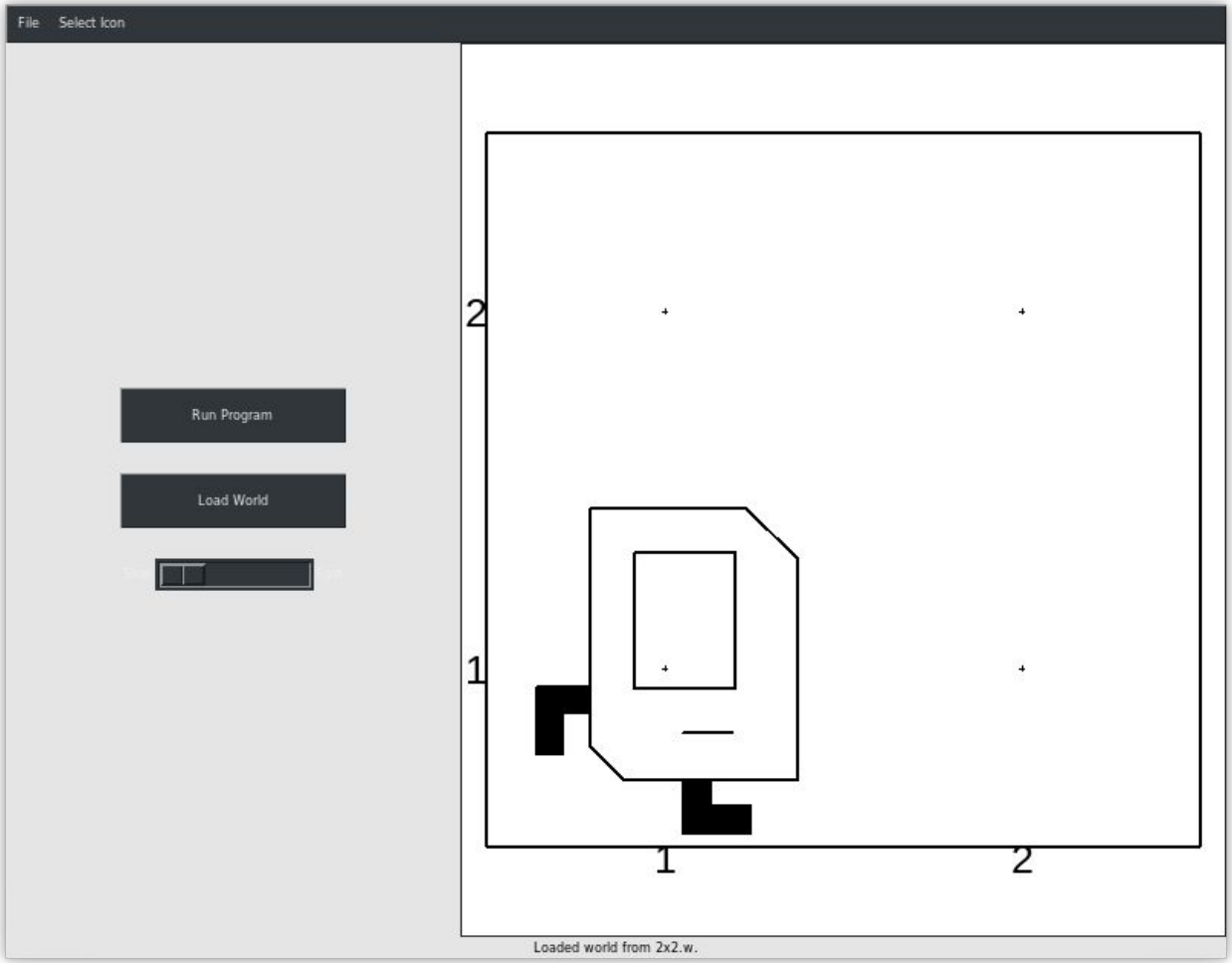
Test	*Opposite*	*What it checks*
`front_is_clear()`	`front_is_blocked()`	Is there a wall in front of Karel?
`left_is_clear()`	`left_is_blocked()`	Is there a wall to Karel's left?
`right_is_clear()`	`right_is_blocked()`	Is there a wall to Karel's right?
`beepers_present()`	`no_beepers_present()`	Are there beepers on this corner?
`beepers_in_bag()`	`no_beepers_in_bag()`	Any there beepers in Karel's bag?
`facing_north()`	`not_facing_north()`	Is Karel facing north?
`facing_east()`	`not_facing_east()`	Is Karel facing east?
`facing_south()`	`not_facing_south()`	Is Karel facing south?
`facing_west()`	`not_facing_west()`	Is Karel facing west?


condition
def main():
while front_is_clear():
put_beeper()
move()
put_beeper()
body
indentation
command outside body

```
def main():
    while front_is_clear():
        put_beeper()
        move()
```

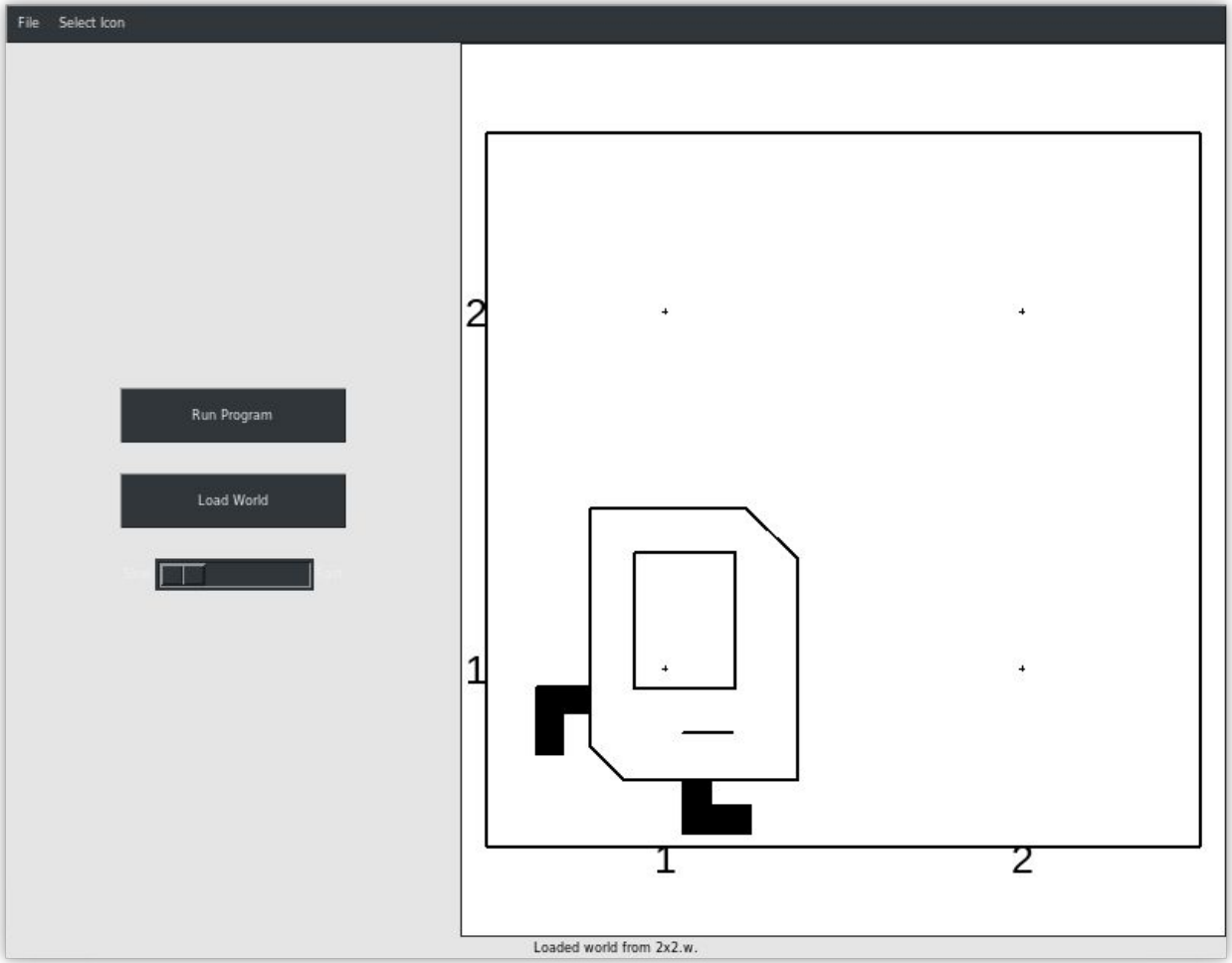


```
def main():
    while front_is_clear():
        put_beeper()
        move()
```

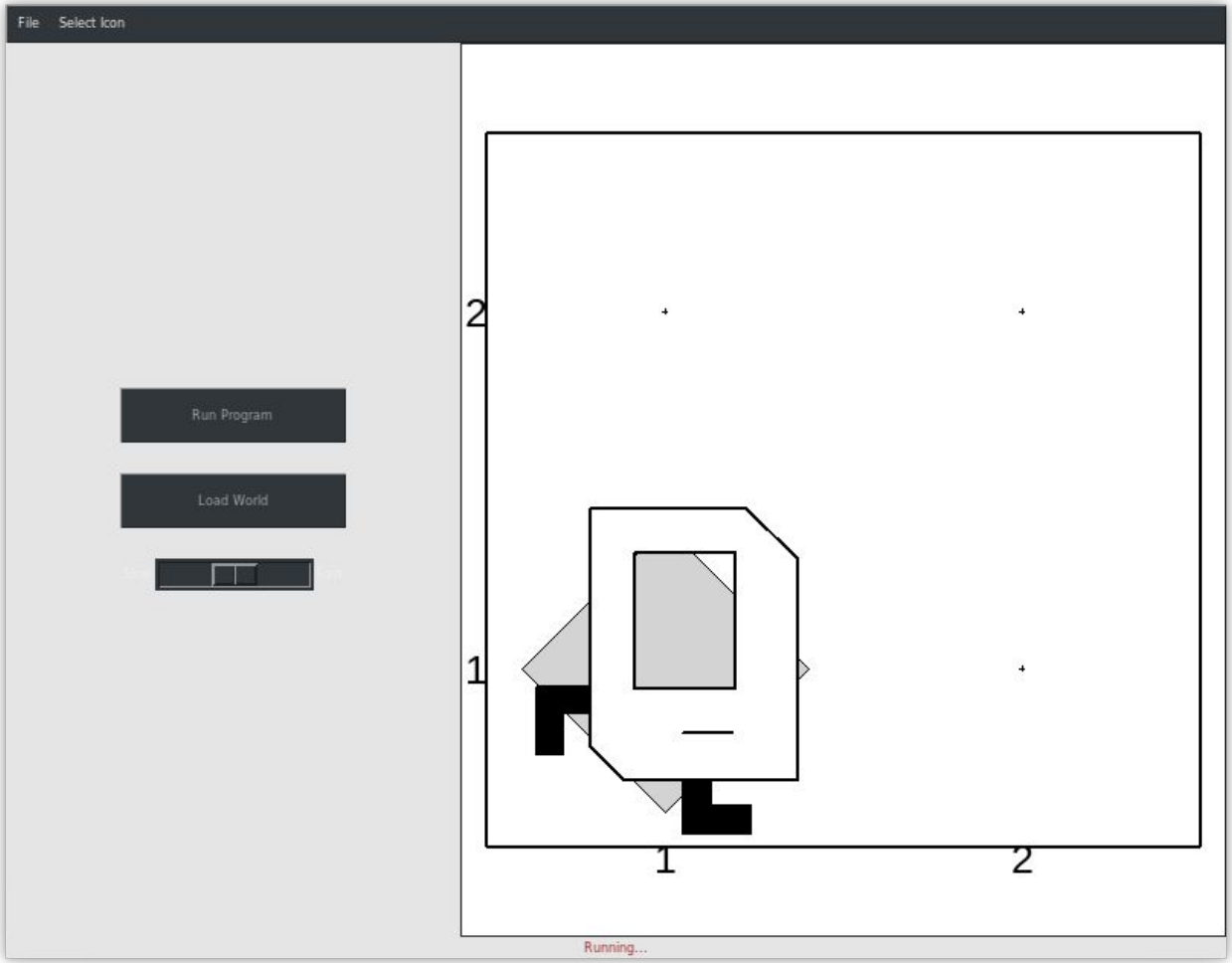


```
def main():
    while front_is_clear():
        put_beeper()
        move()
```

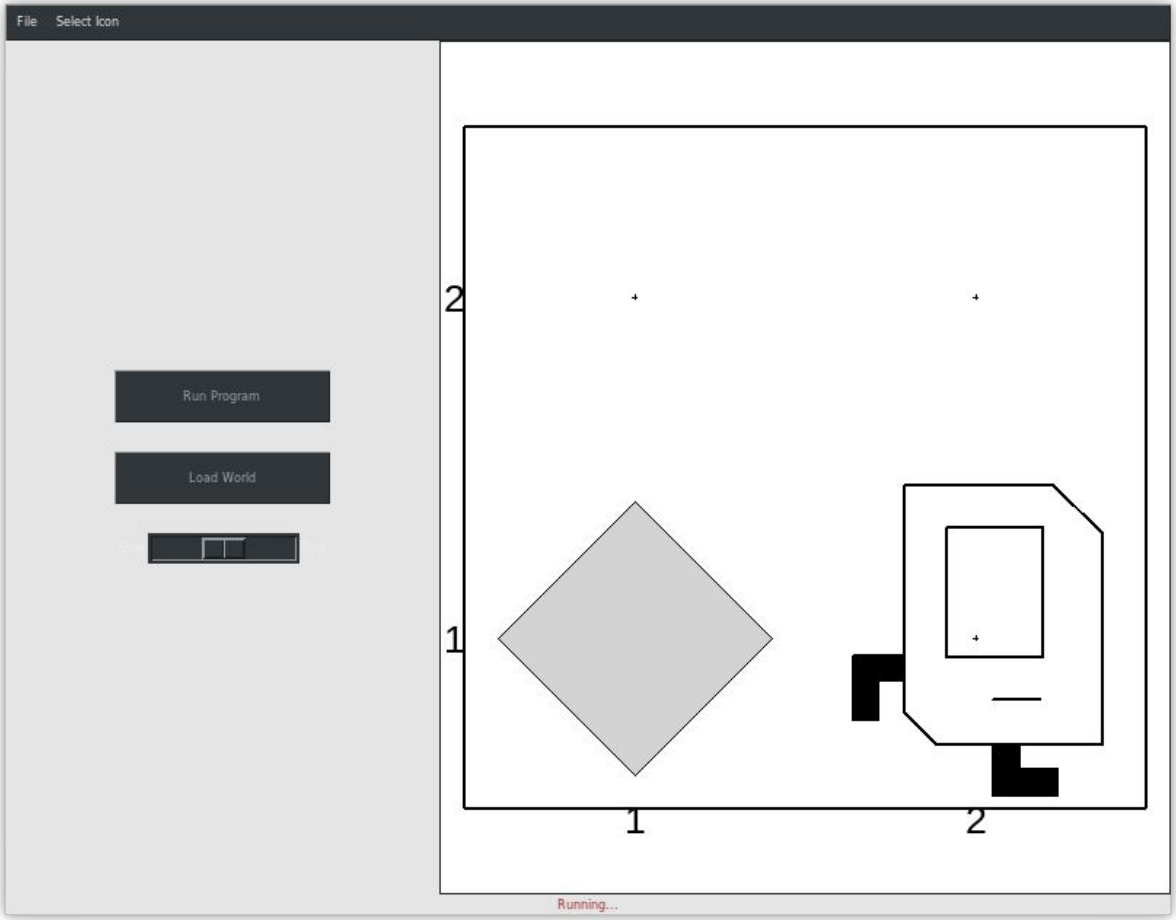
YES



```
def main():
    while front_is_clear():
        put_beeper()
        move()
```

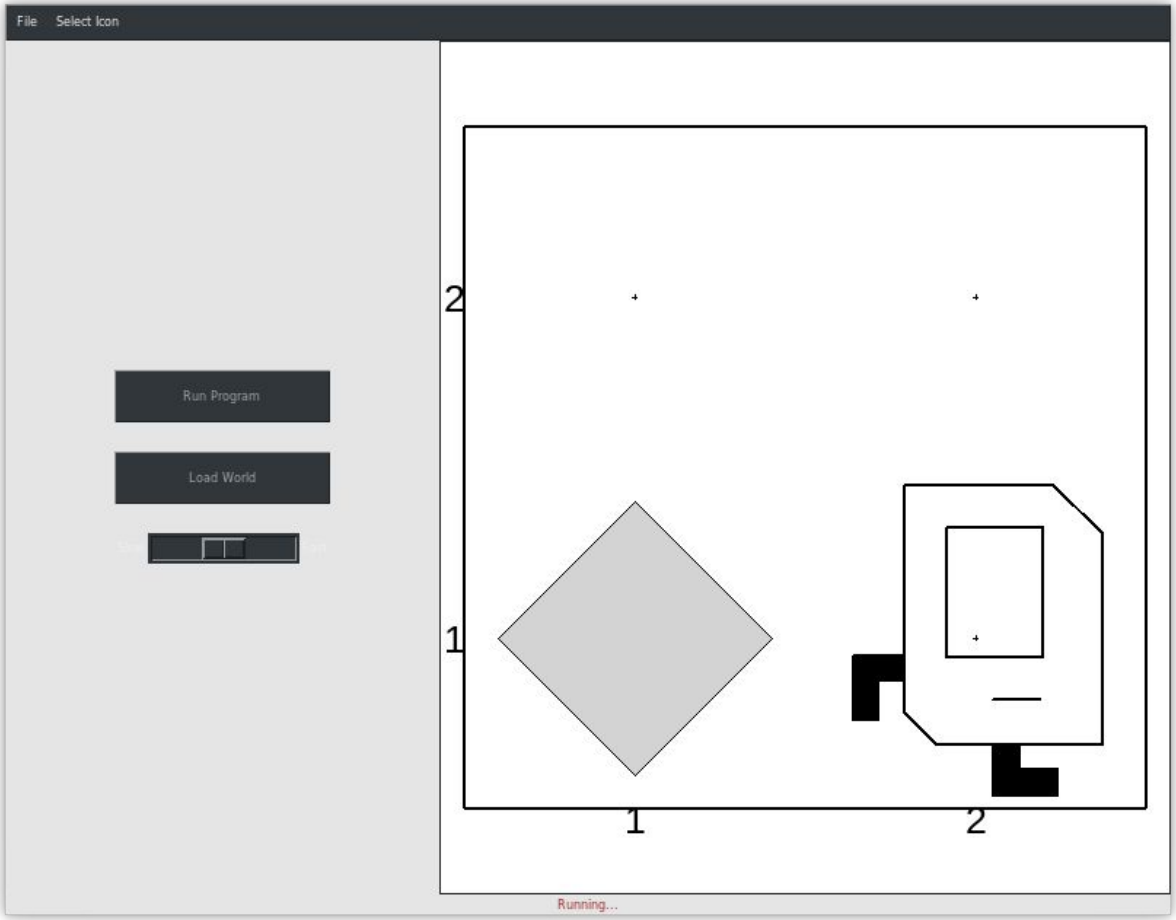


```
def main():
    while front_is_clear():
        put_beeper()
        move()
```

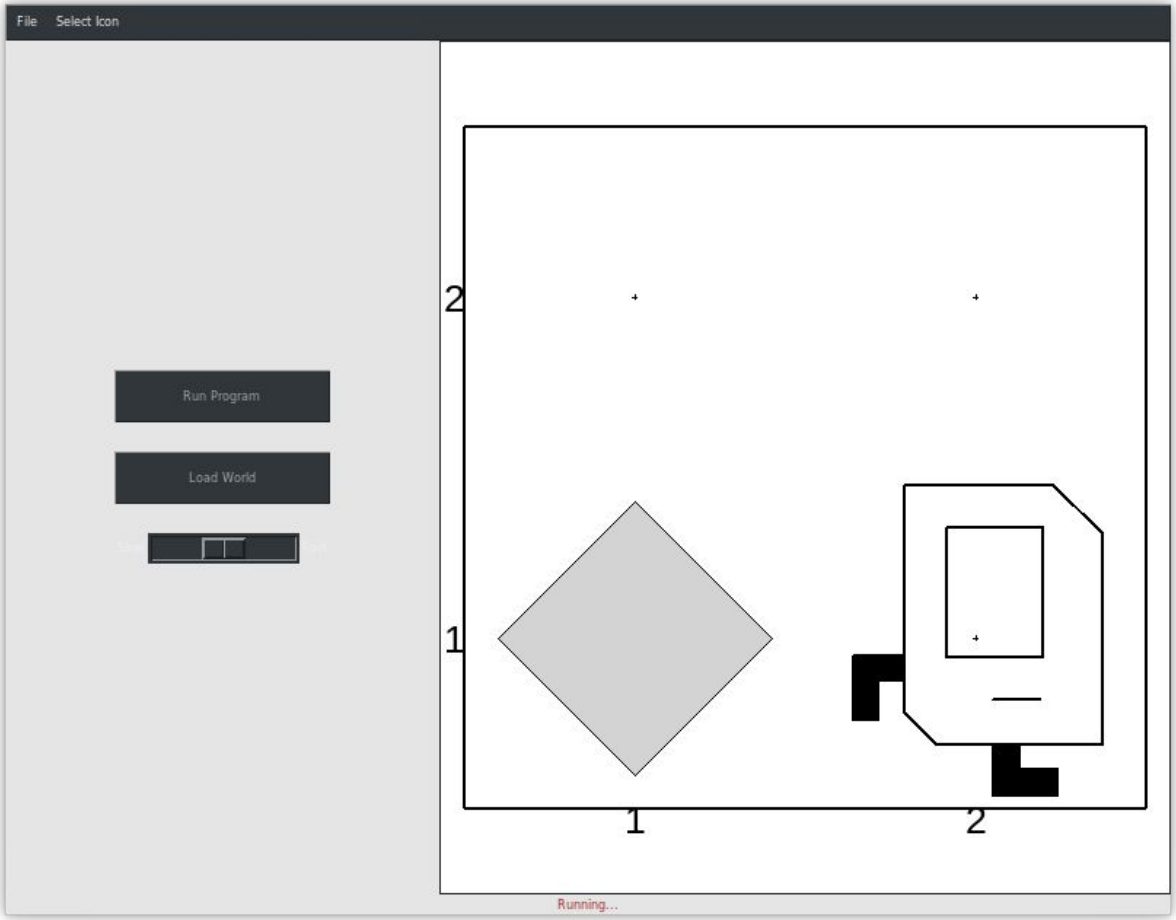


```
def main():
    while front_is_clear():
        put_beeper()
        move()
```

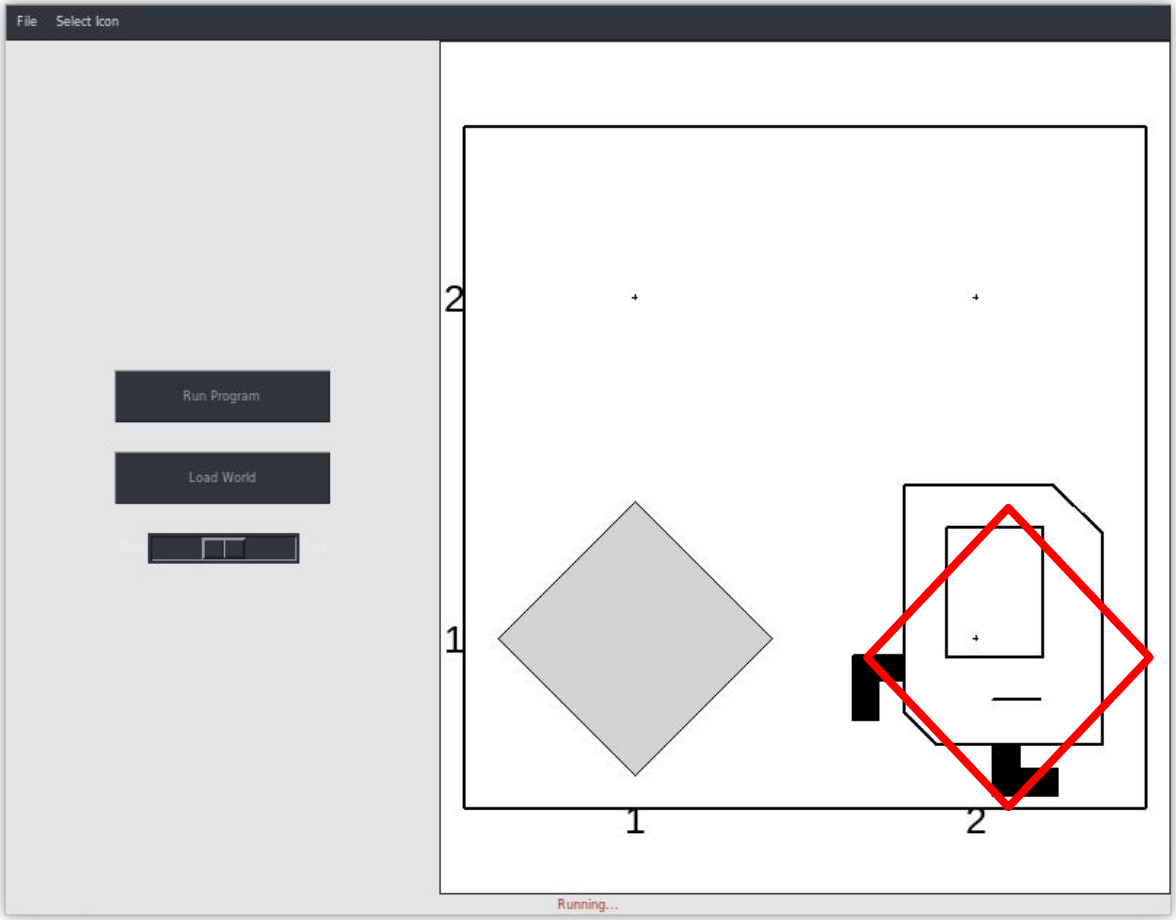
NO



```
def main():
    while front_is_clear():
        put_beeper()
        move()
```



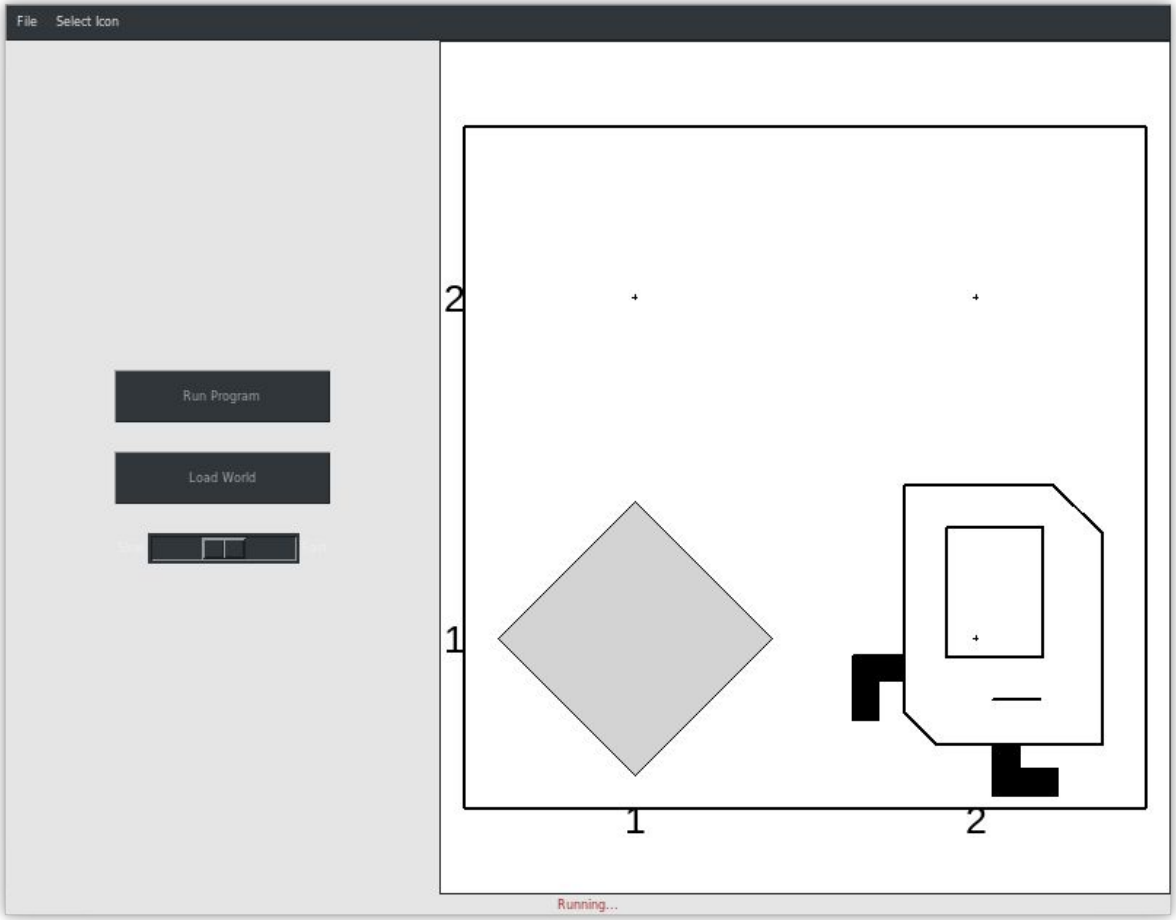
```
def main():
    while front_is_clear():
        put_beeper()
        move()
```



N beepers

N moves

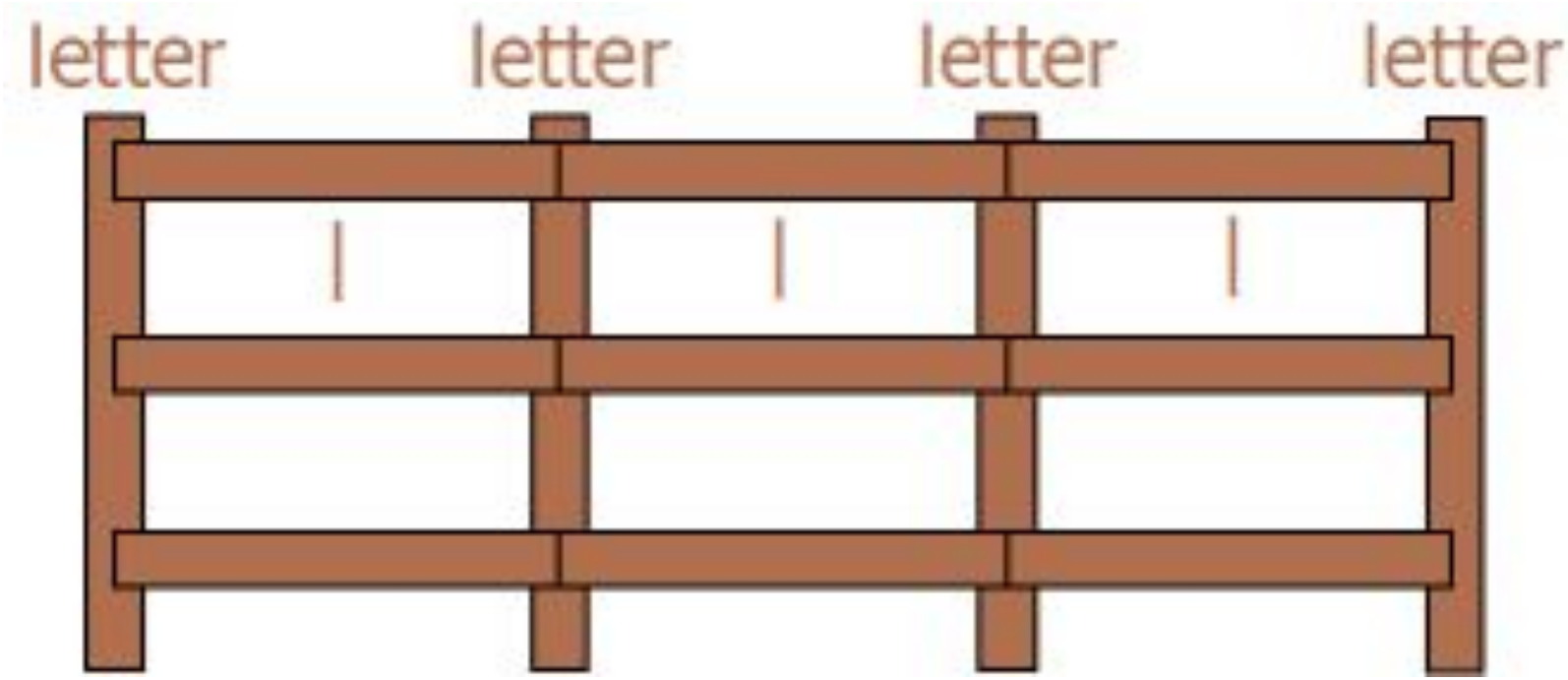
```
def main():
    while front_is_clear():
        put_beeper()
        move()
```



N beepers

N−1 moves

Fence Post Problem

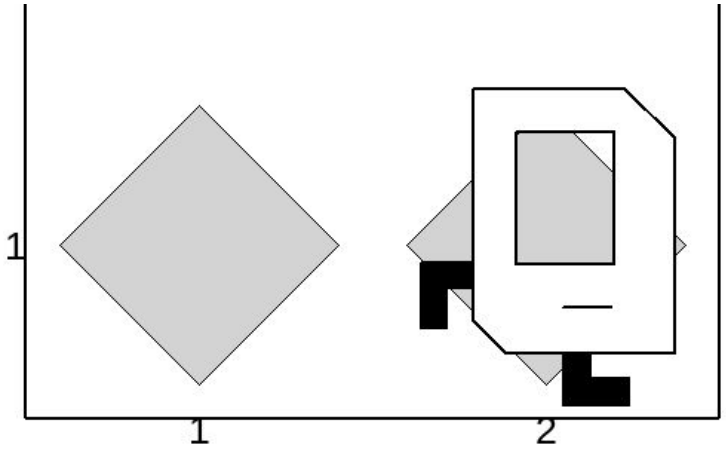


Also sometimes called an “Off By One Error”

```
def main():
    while front_is_clear():
        put_beeper()
        move()
    put_beeper()
```

Happens *N*−1-times

Happens *N*-times (together)



```
for i in range(25):
    put_beeper()
    move()
put_beeper()
```

```
while front_is_clear():
    put_beeper()
    move()
put_beeper()
```